RECOVERING THE MEANING OF DIAGRAMS AND SKETCHES

Alan Mackworth
Department of Computer Science
University of British Columbia
Vancouver, B.C. V6T 1W5

ABSTRACT

Humans exploit diagrams and sketches in everyday communication with each other. Such images convey information because they have meanings fixed by the graphic conventions of the domain. In circuit schematics, architects' plans, program structuring diagrams, stick figures, sketch maps and the like, image structure depicts structure in the scene domain. Computer graphics has traditionally concentrated on representing and manipulating image structures and three-dimensional scene structures. But a friendly computer system must be able to share its interpretation of a diagram with the user. Expert systems that acquire knowledge from computer-naive experts particularly require this capacity. It can only be achieved by explicitly representing the scene/image mapping process. Various explicit representations have been proposed including grammars, constraint methods, predicate calculus and object-oriented schemata. Working systems that use these representations to interpret diagrams and sketches are discussed. The representations are evaluated using descriptive and procedural adequacy criteria. The advantages of explicit knowledge representations for image synthesis and analysis, image transmission and human-machine communication are described.

KEYWORDS: Graphics, Vision, Image-based Systems, Knowledge Representation, Human-Machine Communication, Image Analysis, Image Synthesis, Scene Description.

1. Introduction

Image-based computational systems can be roughly classified as graphics systems and vision systems concerned, respectively, with the synthesis and analysis of images. It is surprising to realize how little common development the fields of graphics and vision have shared since their inception. One purpose of this paper is to point this out, to suggest reasons why it is so and, most importantly, to show that it may now be possible to change that situation.

Although the origin of the two fields can be traced back to two seminal works at MIT in the early 1960's, Sutherland's (1965) Sketchpad system and Roberts' (1965) blocks world vision system, their subsequent development has for the most part occurred in separate institutions, with separate journals, conferences, workshops and textbooks. In short the two communities are sociologically divorced. Moreover, they often have differing purposes. Vision researchers may be involved in artificial intelligence projects that are aimed at understanding the general problem of perception rather than building useful artefacts. For them, the artefacts are the means to that end. For the graphics researcher improved human-machine communication may be the end goal. One could speculate endlessly on the reasons for the lack of a common paradigm. However, there are technical factors underlying the separation, as we shall see.

2. Images Depict Scenes

In any graphics/vision problem we can distinguish an image domain and a scene domain. Images only carry meaning because they depict scenes. For three-dimensional scenes the depiction relation between the scene and the image is specified by the standard imaging process involving illumination conditions, surface reflectance, surface geometry, viewing direction, a centre of projection and occlusion rules. This is well understood in both fields and some of it is even codified in proposed graphics standards (Siggraph, 1979).

What is less well-understood is that all images depict scenes in another domain. Circuit schematics, handwriting, two-dimensional mathematical notation, architects' plans, molecular diagrams, data and control flow diagrams, sketch-maps and the like are all used to communicate information structures graphically. Image domain objects such as lines, regions, junctions and alphanumeric symbols and graphical relationships such as connect, adjacent, parallel, inside, right-of and larger-than are used to communicate

scene domain concepts such as, in a circuit domain, wires, resistors, batteries, current flow and voltage dividers with electrical relationships such as in-series, parallel, common ground and sub-component-of. Human-human and human-machine communication using diagrams and sketches is only successful to the extent that the two communicating entities share a common scene interpretation of the image.

3. Criteria for Judging Knowledge Representations

In graphics the usual way of satisfying the requirement for a shared scene interpretation is to embed the scene domain knowledge, the image domain knowledge and the knowledge of their interrelationship in a specific applications program for a particular scene domain and a particular image domain. Indeed this also has been a prevalent strategy in vision. Since graphics is generally concerned with image synthesis and vision with image analysis the two procedural embeddings of the same knowledge have borne little resemblance to each other and, as a result, there has been little cross-fertilization.

There are many reasons for representing the domain knowledge explicitly rather than implicitly in an opaque, compiled, procedural form, not the least of which is that we may thus be able to share a common computational vision/graphics paradigm. Here two criteria for judging such representations will be given (Stanton, 1972) and then several explicit knowledge representations will be discussed.

The first criterion is descriptive adequacy. Minimally an adequate knowledge representation must explicitly describe the scene domain. There must be a generative set of rules that completely describe the legitimate primitive objects in the domain, their attributes and relationships and any inference rules that allow the formation of new composite objects, attributes or relationships. There must be an analogous set of rules for the image domain. A third necessary component is a complete description of the relation of representation (Clowes, 1971) that describes the image/scene depiction relation. For three-dimensional worlds, for example, the depiction relation is a many-to-one mapping function from the scene domain to the image domain, confounding lighting, surface reflectance, orientation and position, occlusion and image perspective into one or three pixel values (Mackworth, 1983).

Procedural adequacy is the second criterion. The three sets of rules described above must be embodied in a computationally effective procedure. For example, a vision program that interpreted an image by exhaustive analysis-by-synthesis, generating the members of the infinite set of all possible images until one matched the input, would not satisfy this criterion. Standard computational complexity arguments are useful here.

Apart from efficiency, another procedural adequacy consideration is the flexibility of use of the available information. An ideal image-based system would regard all of its potential information sources as inputs or outputs depending on the availability of information. The system would allow control flow from image to scene (image analysis) or from scene to image (image synthesis) bidirectionally or multidirectionally if more than two domains or information sources were available. Or, indeed, information sources may start as partially specified by a symbolic description and the system would refine that description as it used the other information sources and the constraints embedded in the domain knowledge representation. Under this view, the dichotomous classification of potential information sources/sinks as inputs or outputs becomes obsolete. Until we achieve the ideal, we also discuss under procedural adequacy the control facilities provided in the knowledge representation language and the ease of reprogramming the system from synthesis to analysis, say.

The twin criteria of descriptive and procedural adequacy are, of course, often apparently in conflict. The approach usually taken is to favour procedural efficiency, hand-encoding the domain rules into application programs, arguing, "We don't know how to represent, we can't afford and we don't need the full generality of descriptive adequacy". The main purpose of this paper is to show that we are beginning to understand how to achieve descriptive adequacy and procedural adequacy, that we can afford it and that we do need it.

4. Some Knowledge Representations for Vision/Graphics

We can briefly introduce four knowledge representations, point to some programs that have used them and comment on their adequacy.

4.1 Grammars

Chomsky's (1957) paradigm for phrase structure and transformational grammar was intended to satisfy his descriptive adequacy (competence) criteria for natural language understanding. They were generative but supposedly neutral with respect to the procedural (performance) issues of actually producing or understanding language. Many investi-

gators pursued the picture grammar approach as a knowledge representation (e.g. Shaw, 1970). However, from a descriptive point of view the set of relations implicit in a string oriented grammar is inadequate for two-dimensional images and most scene domains. The relation of representation between the image and scene domain is captured not within the grammar but separately using the semantics of the parse tree. Those methods are not well formalized even for string grammars. Procedurally, the efficient parsing strategies for classes of string grammars do not generalize easily to image interpretation, but grammars are neutral with respect to analysis/synthesis. Subsequent work (Browse, 1982) has exploited attribute grammars to enhance the descriptive adequacy. Procedurally, the Augmented Transition Network formalism (Breu & Mackworth, 1980) has allowed the expression of control information to guide interpretation while still allowing analysis and synthesis with the same mechanism.

4.2 Constraints

The Huffman/Clowes/Waltz approach to blocks world vision (Mackworth, 1977a) explicitly represented the image↔scene one↔many mapping. Junctions in the image could depict many differently shaped corners in the scene. Using the rule that an edge must have the same shape at each of its ends, the scene interpretation can be recovered by constraint propagation processes. The Waltz (1972) filtering algorithm is a procedurally efficient way to eliminate local scene ambiguities. It has been generalized to handle general constraint satisfaction problems (Mackworth, 1977b). The paradigm of vision as a constraint satisfaction problem is pervasive (Zucker, 1983). The explicit image/scene mapping is preserved in Mapseel (Mackworth, 1977c), a system for understanding freehand sketch maps. The Marr (1982) Primal Sketch and 2½D sketch assume massively parallel constraint propagation networks as does the Intrinsic Image proposal (Barrow and Tenenbaum, 1978). In that system, the image is registered with a set of intrinsic "images" of scene characteristics such as illumination, albedo, surface orientation and depth that each have internal constraints and constraints with the other images. The computation proceeds by analog constraint propagation, filling in values where none were originally present. This again is neutral with respect to analysis/synthesis: if the intensity image is specified initially the system is doing vision, filling in the other images but if it is not then it produces the image from the other information sources. But the descriptive adequacy is weak in that it allows only pixel-based scalar descriptions not more global, symbolic descriptions (Mackworth, 1983) and it has not been implemented as proposed. Woodham (1980) has successfully demonstrated a technique called photometric stereo that exploits photometric and orientation constraints to interpret image pairs produced by changing the illumination. He has also shown (Woodham, 1983) how to use synthetic images to understand real images thereby integrating image synthesis and analysis.

4.3 Logic

The recent rise of logic programming, and Prolog (Kowalski, 1979) relates to the thesis of this paper. First-order logic is descriptively adequate for vision/graphics tasks. The three sets of generative rules required for typical scene domains such as circuit schematics or human body forms (Browse, 1982) fit nicely into the logic framework. One attractive feature of a Prolog program is its ability to be neutral with respect to the analysis/synthesis question. One problem may be that many implications flow from the scene domain to the image domain but may not be easily reversed. For example, edges parallel in a 3D scene are depicted as parallel lines in the image under orthographic projection but the reverse implication does not follow. Default theories (Reiter, 1980) are required to reverse such implications.

4.4 Schema Representations

Knowledge of the world can often be structured and exploited hierarchically. A composition hierarchy describes each object in terms of its parts, their attributes and relations. Each of the parts is either a primitive object or, recursively, composed of object parts. At the same time, an objects is conveniently described as a specializations of one or more general objects by adding constraints to the description of the more general objects. Such composition and specialization hierarchies are explicitly represented in object-oriented, schema knowledge representation languages.

Mapsee2 (Mackworth and Havens, 1981) uses such a knowledge representation to encode its image and scene knowledge. The program interprets sketch maps of geographical regions depicting landmasses, waterbodies, towns, bridges, lakes, rivers, shores, mountains and the like. The adequacy criteria can be applied to this knowledge representation. The model satisfies the requirements of descriptive adequacy through a natural and complete representational scheme and it is not committed to either analysis or synthesis. Procedural adequacy is, however, ensured by exploiting the properties of the composition and specialization hierarchies for efficient recognition. The programmer can encode top-down, bottom-up or

mixed recognition strategies by attaching procedures to the object schemata. That code is clearly distinct from the object descriptions that are formulated during the recognition process as schema instances. Moreover, of interest here is that the same object representation could be used for graphics, image synthesis and communications applications. Analysis code would be attached to one slot in the object schema while synthesis code would be attached to another slot.

5. Implications for Graphics

What implications are there for graphics in this concern with descriptive and procedural adequacy of knowledge representations for image-based systems?

First of all consider the problem of image transmission. Success has been achieved with image coding in videotext schemes, such as Telidon. Suppose, however, that explicit knowledge representations are established as advocated here. Then instead of transmitting coded image descriptions the sending machine could transmit high-level scene descriptions. If the receiving machine had the same scene, image and scene/image rule base then it could reconstruct the image from the scene description. In many applications orders of magnitude better compression ratios could be achieved using domain-specific coding. Computational vision systems using the knowledge representation would be able to create automatically image and scene-based descriptions from image input without the tedious operator assistance required for Telidon page creation.

For human-machine communication we note that rich communication is only possible with a knowledge representation that allows a shared scene interpretation. Such an interpretation allows dialogue in the scene domain not the image domain. For example, the classic ambiguity of pointing devices can be easily resolved. Free hand sketched input interpreted into the scene domain is more natural than clumsy menu-based drawing systems.

The development of the graphics field has been hampered by the tendency of graphics systems to consist of large, unportable and obscure programs enmeshed in a specific hardware/software environment. Some principles have emerged to alleviate the situation (Newman and Sproull, 1979) such as the use of a high-level language procedural or data structure representation of the image. This encourages device independence and the standardization of graphics interfaces (Siggraph, 1979). The three-way factoring of the knowledge representation advocated here allows us to go beyond device independence to image domain independence. Without changing the scene domain rules one can change the image formatting and object depiction conventions or the very nature of the image domain itself. The argument is that this approach will produce more modular, portable, versatile and intelligent graphics systems with scene domain independence as well.

There is an urgent, practical need for systems like this. For example, expert systems like Prospector (Duda, Gaschnig and Hart, 1979) must acquire rule-based spatial knowledge from a computer-naive expert in the domain being modelled. Currently, a programmer acts as intermediary to uncover the expert's heuristic rules and enter them in coded form. An intelligent graphics systems interfaced to such an expert system would allow direct representation of scene domain configurations in a natural, graphical format. The expert system knowledge acquisition program and the informing expert (or the end user) could communicate in the language of the scene domain - the domain of expertise of the expert system.

6. Conclusion

The rift between graphics and vision is due to the dedication of both fields to emphatically procedural versions of image synthesis and analysis, respectively. By considering new knowledge representation schemes that pay attention to issues of descriptive and procedural adequacy, the rift may soon be bridged. Explicit representations of scene domain, image domain and scene/image mapping knowledge allow flexible use of that knowledge for image synthesis and analysis, image transmission and human-machine communication.

References

Barrow, H.G. and Tenenbaum, J.M. "Recovering Intrinsic Characteristics from Images" in Hanson, A.R. and Riseman, E.M. (eds.) Computer Vision Systems, Academic Press, N.Y. (1978), pp. 3-26.

Breu, H. and Mackworth, A.K. "Push and Pop on Pictures: Generalizing the Augmented Transition Network Formalism to Capture the Structure and Meaning of Images" in Third Nat. Conf. CSCSI, Victoria, B.C., (1980), pp. 172-178.

Browse, R.A. "Knowledge-based Visual Interpretation using Declarative Schemata" Ph.D. Thesis, UBC, Dept. of Computer Science, TR-82-12, (1980).

Chomsky, N. Syntactic Structures, Mouton, The Hague, (1957).

Clowes, M.B. "On Seeing Things" Artificial Intelligence, 2, 1 (1971) 79-112.

Duda, R., Gaschnig, J. and Hart, P. "Model Design in the Prospector Consultant System for Mineral Exploration" in Michie, D. (ed.) Expert Systems in the Microelectronic Age, Edinburgh Univ. Press, (1979).

Kowalski, R. Logic for Problem Solving, North-Holland, Amsterdam, (1979).

Mackworth, A.K. "How to See a Simple World" in Elcock, E.W. and Michie, D. (eds.), Halstead Press, NY (1977a).

Mackworth, A.K. "Consistency in Networks of Relations" Artificial Intelligence 8, 1, (1977b) 99-118.

Mackworth, A.K. "On Reading Sketch Maps", Proc. IJCAI-5, MIT, Cambridge, MA, (1977c), 598-606.

Mackworth, A.K. and Havens, W.S. "Structuring Domain Knowledge for Visual Perception" Proc. IJCAI-7, Univ. of B.C., Vancouver, B.C., (1981).

Mackworth, A.K. "Constraints, Descriptions and Domain Mappings in Computational Vision" in Braddick, O.J. and Sleigh, A.C. (eds.) Physical and Biological Processing of Images, Springer Verlag, Berlin (1983) pp. 33-40.

Marr, D. Vision, W.H. Freeman, San Francisco (1982).

Newman, W.M. and Sproull, R.F. Principles of Interactive Computer Graphics, McGraw-Hill, N.Y. (1979).

Reiter, R. "A Logic for Default Reasoning" Artificial Intelligence 13 (1980) 81-132.

Roberts, L.G. "Machine Perception of Three-Dimensional Objects" in Tippett, J.T. et al (eds.), MIT Press, Cambridge, MA, (1965), pp. 159-197.

Shaw, A.C. "Parsing of Graph Representable Pictures" J.ACM 17, (1970), 453-481.

Siggraph "Status Report of the Graphics Standards Planning Committee of ACM/SIGGRAPH", Computer Graphics, 13, 3 (1979).

Stanton, R.B. "The Interpretation of Graphics and Graphics Languages" in Nake, F. and Rosenfeld, A. (eds.), North-Holland, Amsterdam, (1972) pp. 144-159.

Sutherland, I.E. "Sketchpad: A Man-Machine Graphical Communication System" MIT Lincoln Lab. Tech. Rep. 296, Cambridge, MA, (1965).

Waltz, D. "Understanding Line Drawings of Scenes with Shadows" in Winston, P.H. (ed.) The Psychology of Computer Vision, McGraw-Hill, NY, (1972), pp. 19-91.

Woodham, R.J. "Using Digital Terrain Data to Model Image Formation in Remote Sensing" Proc. SPIE, 238 (1980), pp. 361-369.

Woodham, R.J. "Viewer-Centred Intensity Computations" in Braddick, O.J. and Sleigh, A.C. (eds.) Physical and Biological Processing of Images, Springer-Verlag, Berlin (1983) pp. 217-229.

Zucker, S. "Cooperative Grouping and Early Orientation Selection" in Braddick, O.J. and Sleigh, A.C. (eds.) Physical and Biological Processing of Images, Springer-Verlag, Berlin (1983) pp. 326-334.